How to Install FSindo

Kiyoshi Yagi
kiyoshi.yagi@riken.jp

Theoretical Molecular Science Laboratory
RIKEN Cluster for Pioneering Research

2019/11/12

NOTE

FSindo is a command line based program. This manual assumes that you are familiar with the commands in UNIX, and that you are working on Bourne Shell (bash).

0. Prepare LAPACK and BLAS libraries

- NETLIB: http://www.netlib.org/lapack/
- Intel Math Kernel Library

1. Download sindo-4.0.tar.gz from our website

   https://tms.riken.jp/en/research/software/sindo/

2. Extract the tarball and configure

```
> tar –zxvf sindo-4.0.tar.gz
> cd sindo-4.0/FSindo
> ./configure
(See the next page)
```

3. Build

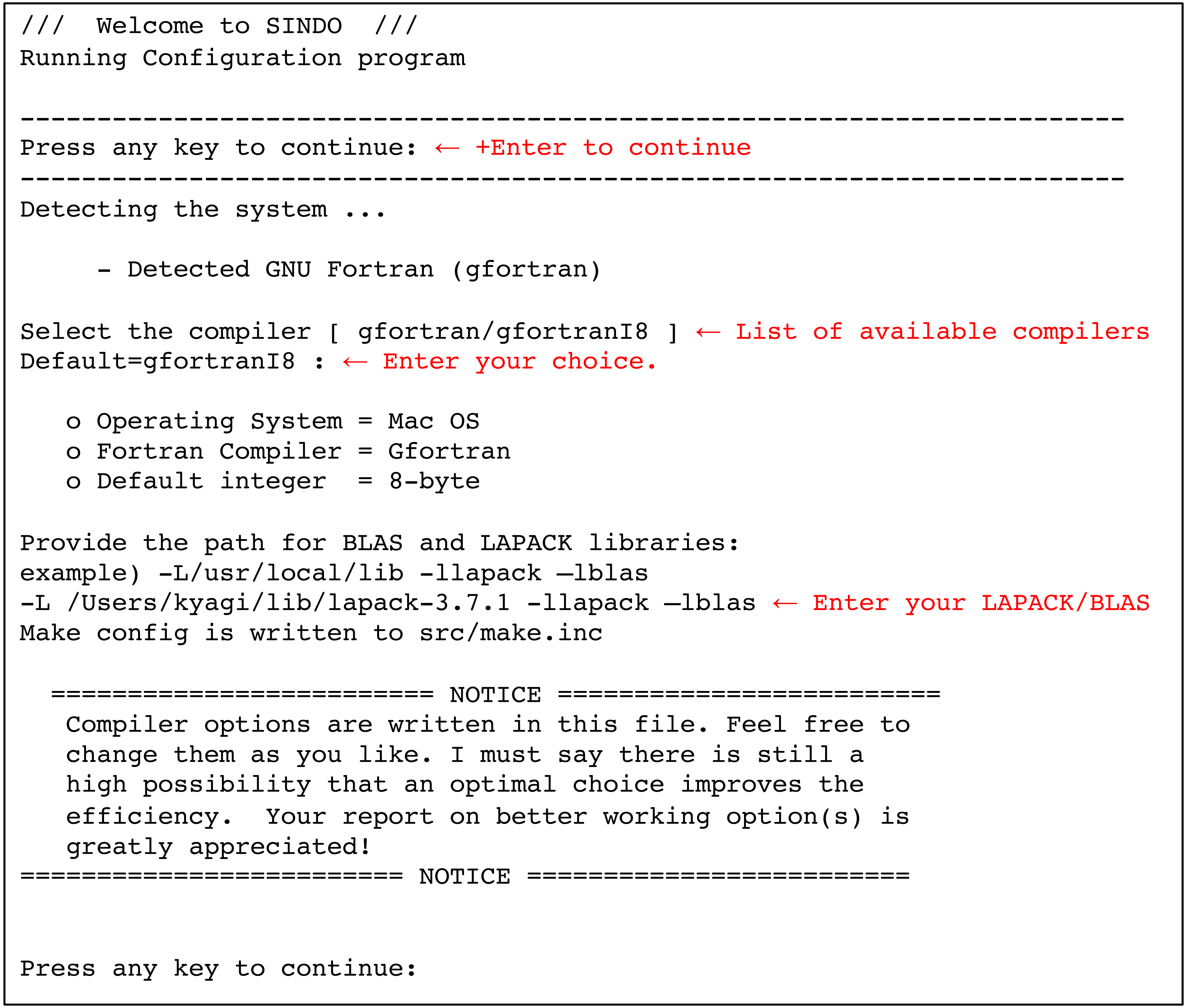
```
> cd src
> make >& make.log
```

- When successful, an executable file will be created in “FSindo/bin/sindo”.

4. Set a path

```
> export PATH=$PATH:/path/to/sindo-4.0/FSindo/bin
```

- It is recommended to write this line in your login script (~/.bashrc).



```
///  Welcome to SINDO  ///
Running Configuration program

------------------------------------------------------------------------
Press any key to continue: ← +Enter to continue
------------------------------------------------------------------------
Detecting the system ...

     - Detected GNU Fortran (gfortran)

Select the compiler [ gfortran/gfortranI8 ] ← List of available compilers
Default=gfortranI8 : ← Enter your choice.

   o Operating System = Mac OS
   o Fortran Compiler = Gfortran
   o Default integer  = 8-byte

Provide the path for BLAS and LAPACK libraries:
example) -L/usr/local/lib -llapack –lblas
-L /Users/kyagi/lib/lapack-3.7.1 -llapack –lblas ← Enter your LAPACK/BLAS
Make config is written to src/make.inc

  ========================= NOTICE =========================
   Compiler options are written in this file. Feel free to
   change them as you like. I must say there is still a
   high possibility that an optimal choice improves the
   efficiency.  Your report on better working option(s) is
   greatly appreciated!
========================= NOTICE =========================


Press any key to continue:
```

If the compilation failed, please inspect “src/make.inc”. Many problems come from wrong path for lapack/blas libs (“LAPACK”) and/or from fortran options (F90OPT, F77OPT).

Example for gfortran/netlib

```
SINDO_ROOT = /Users/kyagi/Work/devel/sindo/sindo.master/Fsindo
TARGET = gfortranI8
LAPACK = -L/Users/kyagi/Work/lib/lapack-3.7.1 -llapack -lblas
RM = rm
```

⋮

```
# Fortran77 compiler & option with and without optimization
F77C = gfortran
F77OPT= -fdefault-integer-8 -O2 -funroll-loops -fomit-frame-pointer
F77NOOPT= -fdefault-integer-8 -O0

# Fortran90 compiler & option with and without optimization
F90C = gfortran
F90OPT= -fdefault-integer-8 -O2 -funroll-loops -fomit-frame-pointer
F90NOOPT= -fdefault-integer-8 -O0
```

Example for intel/MKL

```
SINDO_ROOT = /Users/kyagi/Work/devel/sindo/sindo.master/Fsindo
TARGET = ifortI8_MKL_sequential
LAPACK = $(MKLROOT)/lib/intel64/libmkl_blas95_ilp64.a ¥
                       $(MKLROOT)/lib/intel64/libmkl_lapack95_ilp64.a ¥
                       -Wl,--start-group ¥
                         $(MKLROOT)/lib/intel64/libmkl_intel_ilp64.a ¥
                         $(MKLROOT)/lib/intel64/libmkl_sequential.a ¥
                         $(MKLROOT)/lib/intel64/libmkl_core.a ¥
                       -Wl,--end-group ¥
                       -lpthread –lm
RM = rm
```

⋮

```
# Fortran77 compiler & option with and without optimization
F77C = ifort
F77OPT= -i8 -w -cm -static -O3 -funroll-loops
F77NOOPT= -fdefault-integer-8 -O0

# Fortran90 compiler & option with and without optimization
F90C = ifort
F90OPT= -i8 -w -cm -static -O3 -funroll-loops
F90NOOPT= -fdefault-integer-8 -O0
```